



Technical Newsletter

File Number S360-48
Re: Form No. C28-6539-4
This Newsletter No. N28-2214
Date February 27, 1967
Previous Newsletter Nos. None

IBM SYSTEM/360 OPERATING SYSTEM JOB CONTROL LANGUAGE

This technical newsletter amends the publication, IBM System/360 Operating System: Job Control Language, Form C28-6539-4. Corrections and additions to the text are noted by vertical bars at the left of the change.

<u>Pages to Be</u> <u>Inserted</u>	<u>Pages to Be</u> <u>Removed</u>
19,20	19,20
29,30	29,30
47-50	47-50
53,54	53,54
57,58	57,58
67,68	67,68
73,74	73,74
81,82	81,82

Summary of Amendments

Documentation of the SYSUDUMP DD statement, bypass label processing, and the password protection feature. Additional unit types are included in Appendix A.

Note: Please file this cover letter at the back of the publication. Cover letters provide a quick reference to changes and a means of checking receipt of all amendments.

Setting Job Step Time Limits (TIME)
(Priority Schedulers Only)

To limit the computing time used by a single job step or cataloged procedure step, you might want to assign a maximum time for its completion. Such an assignment is useful in a multiprogramming environment where more than one job has access to the computing system.

To assign a time limit to a job step, code the keyword parameter

TIME=(minutes,seconds)

in the operand field of the EXEC statement. Replace the terms "minutes" and "seconds" with the maximum number of minutes and seconds allotted for execution of the job step. The number of minutes cannot exceed 1439; the number of seconds cannot exceed 59. If the job step is not completed in this time, the entire job is terminated.

If you omit the TIME parameter, the default job step time limit (as established in the cataloged procedure for the input reader) is assumed. If the job step execution time may exceed 1439 minutes (24 hours), code TIME=1440 to eliminate job step timing.

Notes:

- If the time limit is given in minutes only, you need not code the parentheses, e.g., TIME=5.
- If the time limit is given in seconds only, you must code a comma to indicate the absence of minutes, e.g., TIME=(,45).
- When the job step uses a cataloged procedure, you can set a time limit for a single procedure step by including, as part of the keyword TIME, the procedure step name, i.e., TIME.procstepname. This specification overrides the TIME parameter in the named procedure step, if one is present. You can code as many parameters of this form as there are steps in the cataloged procedure.
- To set a time limit for an entire procedure, code the TIME parameter without a procedure step name. This specification overrides all TIME parameters in the procedure, if any are present.

Specifying Main Storage Requirements for a Job Step (REGION)
(Priority Schedulers Only)

For job steps that require an unusual amount of main storage, the EXEC statement provides you with the REGION parameter. Through this parameter you can specify the maximum amount of main storage to be allocated to the associated job step. This size must take into account the system components required by your installation.

To specify a region size, code the keyword parameter

REGION=nnnnnK

in the operand field of the EXEC statement. Replace the term "nnnnn" with the number of 1024-byte areas you want allocated to the job step, e.g., REGION=51K. This number can range from one to five digits.

If you omit the REGION parameter, the default region size (as established in the cataloged procedure for the input reader) is assumed.

Notes:

- If you have specified a REGION parameter in the JOB statement, REGION parameters in the job's EXEC statements are ignored.
- When the job step uses a cataloged procedure, you can request a region size for a single procedure step by including, as part of the REGION parameter, the procedure step name, i.e., REGION.procstepname. This specification overrides the REGION parameter in the named procedure step, if one is present. You can code as many parameters of this form as there are steps in the cataloged procedure.
- To request a single region size for an entire cataloged procedure, code the REGION parameter without a procedure step name. This specification overrides all REGION parameters in the procedure, if any are present.

Reference:

- The storage requirements you must consider when specifying a region size are outlined in the publication IBM System/360 Operating System: Storage Estimates.

DD STATEMENT

Data sets used by processing programs must be represented by DD statements in the input stream. The DD statements pertaining to a particular job step follow the EXEC statement associated with the step. A DD statement must contain the term DD in its operation field.

The DD statement is the final source of information that is needed to retrieve and store data. Figure 2 illustrates the sources of information and the means by which each source refers to the next.

An input/output macro-instruction, coded as part of the processing program, issues an input or output request (OPEN, CLOSE, GET, PUT, READ, WRITE). This request uses a dcbname to refer to a data control block created earlier by a DCB macro-instruction. The data control block contains information about a data set that is gathered from several sources, one of which is a DD statement whose ddname matches the ddname given in the data control block. The DD statement, the final source of information, is associated with a specific data set. It refers to the data set with a data set name.

Because of the DD statement's position in this sequence of information sources, you can specify such characteristics as buffer size, record length, and device type at the time the job step is executed, rather than when you code the processing program.

To follow the flow of parameters in the DD statement, turn to Appendix E and fold

out Chart 2 while reading this chapter. Individual parameters are shown in detail in a series of figures on Chart 3 of Appendix E.

Identifying the DD Statement (ddname)

The ddname identifies the DD statement so that subsequent control statements and the data control block can refer to it. It must satisfy the position, length, and content requirements for a name field. Each ddname within a job step should be unique. If duplicate ddnames exist, all references are directed to the first such DD statement in the job step, and the second is ignored.

Note:

- Omit the ddname if the data set is concatenated with the data set defined by the preceding DD statement, or the DD statement is one of a group of DD statements that define an indexed sequential data set.

If the job step uses a cataloged procedure, the ddname must be qualified by the procedure step name, i.e., procstepname.ddname. The ddname can identify either a DD statement in the procedure, whose parameters you want to override, or a new DD statement you want to add to the procedure. In both cases, the modification is valid only for the duration of the job step; it does not change the procedure permanently.

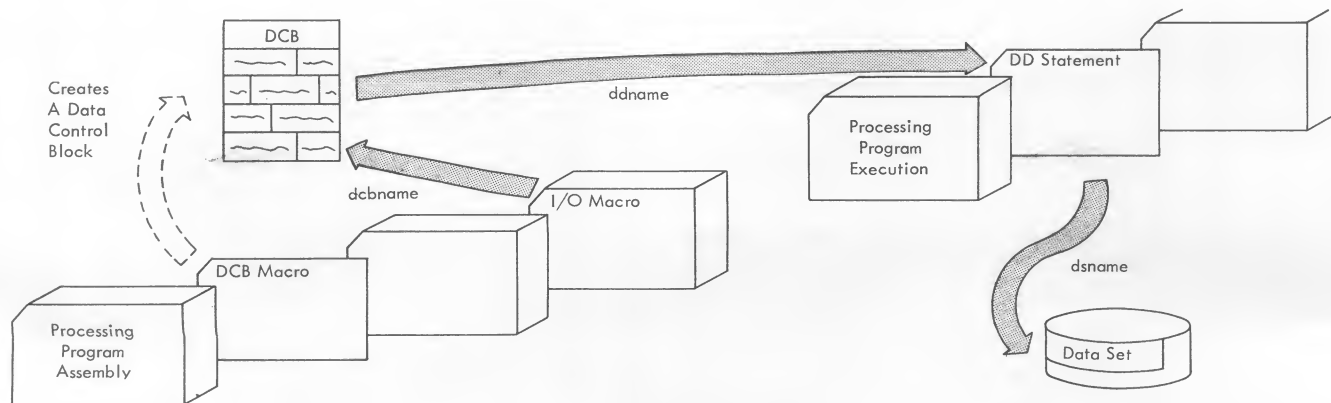


Figure 2. Data Set Information Sources

in the operand field of the DD statement. Replace the word "dsname" with the data set's cataloged name. The volume that contains this data set must be mounted before the execution of the job step containing the copy request. A permanently resident volume is the most likely place from which to copy such information, in that it is always mounted.

If such a data set does not exist, you still might be able to copy the DCB parameter of an earlier DD statement in the job. To refer to this DD statement, code the keyword parameter

DCB=*.stepname.ddname

in the operand field. Replace the terms "stepname" and "ddname" with the job step name and DD statement name, respectively.

Notes:

- If the earlier DD statement is contained in the same job step, you need not code the stepname, i.e., DCB=*.ddname.
- If the earlier DD statement is contained in a cataloged procedure step, you must include the procedure step name, i.e., DCB=*.stepname.procstepname.ddname.

If you wish to modify the information that is copied from another data set label or DCB parameter, code

DCB=(reference,list of attributes)

Replace the term "reference" with dsname or *.stepname.ddname. The attributes in the list override the corresponding copied attributes. Data set attributes are coded in the form of keyword subparameters separated by commas, e.g., BLKSIZE=810 for a block size of 810 bytes. These subparameters correspond to operands in the DCB macro-instruction and are coded using the same keywords and values. A glossary of valid DCB subparameters is given in Appendix B of this publication.

If you cannot copy another data set label or DCB parameter, you must supply all DCB attributes that are not specified in the processing program (either directly or by default) or data set label. Code the keyword parameter

DCB=(list of attributes)

in the operand. Again, the attributes are coded as keyword subparameters separated by commas, e.g., DCB=(RECFM=FB,LRECL=80,...).

References:

- DCB macro-instructions and operands are described in detail in the publication IBM System/360 Operating System: Supervisor and Data Management Macro-Instructions.
- DCB macro-instructions and operands associated with the graphic access methods are described in the publications IBM System/360 Operating System: Graphic Programming Services for the IBM 2250 Display Unit, Form C27-6909, and Graphic Programming Services for IBM 2260 Display Station (Local Attachment), Form C27-6912.

Describing the Data Set Label (LABEL)

Data sets residing on magnetic tape volumes usually have data set labels. Those on direct-access volumes must have labels conforming to standard label specifications. The LABEL parameter indicates the label type, the data set's relative position on tape, its retention period, and whether a password is required to read or write on it. To supplement this discussion of the LABEL parameter pictorially, turn to Appendix E and fold out Chart 3.

Magnetic tape volumes can contain volume labels and data set header and trailer labels that do not conform to the system standard label specifications. To create or retrieve a data set residing on such a tape volume, you must include the LABEL parameter. To specify the label type, code

LABEL=(,type)

in the operand field. Replace the word "type" with:

- SL - if the data set has standard labels.
- NL - if the data set has no labels.
- NSL - if the data set has nonstandard labels.
- SUL - if the data set has both standard and user labels.
- BLP - to bypass label processing.

If you specify SUL, SL, or omit the label type (in which case standard labels are assumed), the operating system will ensure that the correct volumes are mount-

ed. If you specify NSL, your installation must have incorporated label processing routines into the operating system. If you specify NL, the data set must have no labels.

The feature that allows you to bypass label processing is a system generation option (OPTIONS=BYLABEL). If this option was not requested at system generation and you have coded BLP, the system assumes NL.

Note:

- When BLP is specified, you should ensure that the operator mounts the correct tape volume before processing it.

If the data set is not first in sequence on the reel, the LABEL parameter serves to position the tape properly through a data set sequence number. Code

```
LABEL=seq#
```

in the operand field. Replace the term "seq#" with the 1- to 4-digit sequence number assigned to the data set when it was created.

Note:

- If 0 appears as the data set sequence number, the system assumes 1.

Both magnetic tape and direct-access data sets can be assigned a retention period and password protection when they are created. If you wish the data set to remain intact for some period of time, you

can specify either the length of time in days or the exact date you want it to expire. Otherwise, a retention period of zero days is assumed. After expiration, the data set can be deleted, or opened for any type of output. To specify a retention period, code

```
LABEL=RETPD=nnnn
```

in the operand field. Replace the term "nnnn" with the number of days you want the data retained. To specify, instead, an expiration date, code

```
LABEL=EXPDT=yyddd
```

in the operand field. Replace the term "yyddd" with the 2-digit year number and 3-digit day number after which the data set can be considered expired.

If you wish the data set to be accessible only through the use of a password, code

```
LABEL=(, ,PASSWORD)
```

in the operand field. The operating system assigns the data set security protection. To retrieve it, the operator must respond to a message by issuing the correct password.

Note:

- Subparameters in the LABEL parameter can be coded in various combinations. The terms seq#, type, and PASSWORD are all positional subparameters.

Defining Job Step Boundaries

Job step boundaries are ordinarily established by EXEC statements. Each EXEC statement marks the beginning of a new job step and the completion of control statements for the previous job step. The completion of the last step in a job is marked with a JOB statement associated with the succeeding job, or a null statement. Figure 4 shows a group of job steps and their boundaries.

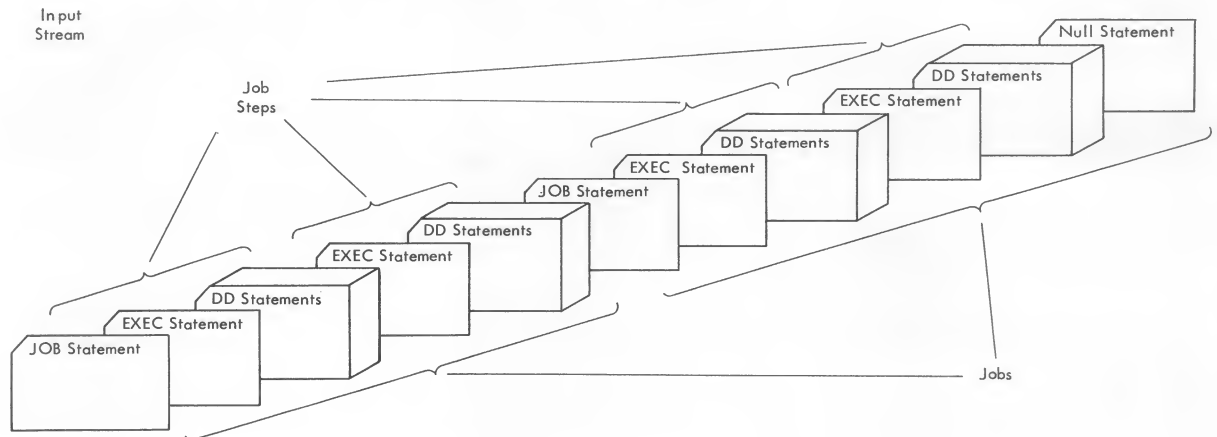


Figure 4. Defining Job Step Boundaries

CREATING NEW OUTPUT DATA SETS

Output data sets in a job step that do not exist before the step is executed are created by using subsets of the DD statement parameters. The contents of each subset depend primarily on what type of device the data set resides on.

Creating Unit Record Data Sets

Data sets whose destination is a printer or card punch are created with the DD statement parameters UNIT and DCB.

UNIT: Required. Code unit information using the 3-digit address (e.g., UNIT=00E), the type (e.g., UNIT=1403), or the system-generated group name (e.g., UNIT=PRINTER).

DCB: Required only if the data control block is not completed in the processing program. Valid DCB subparameters are listed in Appendix B.

Examples 1 and 2 illustrate valid DD statements for creating data sets on the printer or card punch.

```
[/OUTPUT DD UNIT=1403
```

Example 1. Creating a Data Set on the Printer

```
[/OUTPUT DD UNIT=2520,DCB=(STACK=2,MODE=E)
```

Example 2. Creating a Data Set on a Card Punch

Creating Data Sets on Magnetic Tape

Tape data sets are created using combinations of the DD statement parameters UNIT, LABEL, DSNAME, DCB, VOLUME, and DISP.

UNIT: Required, except when volumes are requested using VOLUME=REF. You can assign a unit by specifying its address, type, or group name, or by requesting unit affinity with an earlier data set. You can also request multiple output units and defer volume mounting with this parameter.

LABEL: Required when the tape does not have standard labels, and when the data set does not reside first on the reel. It is also used to assign a retention period and password protection.

DSNAME: Required for data sets that are to be cataloged or used by a later job.

DCB: Required only when data control block information is not completely specified in the processing program. Usually, such attributes as the logical record length (LRECL) and buffering technique (BFTEK) will have been specified in the processing program. Other attributes, such as the maximum block size (BLKSIZE) and the tape recording technique (TRTCH), are more appropriately specified in the DD statement. Valid DCB subparameters are listed in Appendix B.

VOLUME: Optional, you can use this parameter to request specific volumes. If you use VOLUME=REF, and want to save existing data sets on the specified volume(s), you must indicate the data set sequence number in the LABEL parameter.

DISP: Required for data sets that are to be cataloged, passed, or kept.

Examples 3 through 8 illustrate valid DD statements for creating data sets on magnetic tape.

```
//OUTPUT2 DD UNIT=2400-2
```

Example 3. Creating a Temporary Data Set on Labeled Tape

```
//OUTPUT2 DD UNIT=2400,LABEL=(,NL)
```

Example 4. Creating a Temporary Data Set on Unlabeled Tape

```
//OUTPUT2 DD UNIT=2400-2,DSNAME=A.B.C,DISP=(,CATLG),  
// LABEL=RETPD=0090
```

Example 5. Creating and Cataloging a Tape Data Set

```
//OUTPUT2 DD DSNAME=&WORK,VOLUME=REF=*.STEP1.OUTPUT,  
// DISP=(,PASS),LABEL=(,NL)
```

Example 6. Creating a Temporary Data Set on Unlabeled Tape, Using VOLUME=REF

```
//OUTPUT DD DSNAME=ALPHA,UNIT=2400,DISP=(,KEEP),LABEL=2,  
// VOLUME=SER=T2
```

Example 7. Creating and Keeping a Data Set Second in Sequence on a Labeled Tape

```
//OUTPUT2 DD UNIT=2400-2,DCB=(TRTCH=C,LRECL=256,DEN=1)
```

Example 8. Creating a Temporary Data Set Having an Incomplete Data Control Block

Creating Sequential (BSAM or QSAM) Data Sets on Direct-Access Devices

Sequential data sets are created using combinations of the DD statement parameters UNIT, DSNAME, VOLUME, LABEL, DISP, DCB, and one of the space allocation parameters SPACE, SPLIT, or SUBALLOC.

UNIT: Required, except when volumes are requested using VOLUME=REF or space is allocated using SPLIT or SUBALLOC. You can assign a unit by specifying its address, type, or group name, or by requesting unit affinity.

DSNAME: Required for all but temporary data sets.

LABEL: Required if you want to assign a retention period or password protection.

DCB: Required only when data control block information is not completely specified in the processing program. Usually, such attributes as the logical record length (LRECL) and buffering technique (BFTEK) will have been specified in the processing program. Other attributes, such as the maximum block size (BLKSIZE) and the number of buffers (BUFNO) are more appropriately specified in the DD statement. Valid DCB subparameters are listed in Appendix B.

VOLUME: Optional. You can use this parameter to request specific volumes, multiple nonspecific volumes, and to specify PRIVATE and RETAIN.

DISP: Required for data sets that are to be cataloged, passed, or kept.

SPACE, SPLIT, SUBALLOC: One of these is required for all new direct-access data sets.

Examples 9 through 14 illustrate valid DD statements for creating sequential data sets on direct-access devices.

```
//OUTPUT3 DD UNIT=2311,SPACE=(TRK,(10,2))
```

Example 9. Creating a Temporary Data Set on Disk

```
//OUTPUT3 DD DSNAME=6BUF,DISP=(,PASS),SPLIT=10
```

Example 10. Creating a Temporary Disk Data Set That Shares Cylinder Space With the Preceding Data Set

```
//OUTPUT3 DD UNIT=2301,DCB=(BLKSIZE=1026,RECFM=FB), X
// SPACE=(1026,(30,3),,CONTIG,ROUND)
```

Example 11. Creating a Temporary Drum Data Set, With Space Allocation in Blocks

```
//OUTPUT3 DD SUBALLOC=(TRK,(5,1),STEP1.OUTPUT)
```

Example 12. Creating a Temporary Disk Data Set, Using Suballocation Technique

```
//OUTPUT3 DD DSNAME=ALPHA,UNIT=2311,DISP=(,KEEP), X
// SPACE=(1024,(200,10)),VOLUME=(PRIVATE,RETAIN,SER=D02)
```

Example 13. Creating and Keeping a Data Set on a Private Disk Pack

```
//OUTPUT3 DD DSNAME=X.Y.Z,VOLUME=REF=*.STEP1.OUTPUT,DISP=(,CATLG), X
// SPACE=(CYL,(2,1)),LABEL=EXPDT=67365
```

Example 14. Creating and Cataloging a Disk Data Set, Using VOLUME=REF

RETRIEVING EXISTING DATA SETS

Data sets that exist before the job step that uses them are retrieved using subsets of the DD statement parameters. The contents of each subset depend on the disposition assigned to the data set when it was created. Existing data sets can be used for both input and output purposes. In the latter case, the existing data set is extended with additional output instead of being read as input.

Retrieving Cataloged Data Sets

Input data sets that were assigned a disposition of CATLG, or were cataloged by the IEHPROGM utility program, are retrieved using the DD statement parameters DSNAME, DISP, LABEL, and DCB. The device type, volume serial number, and data set sequence number (if tape) are stored in the catalog.

DSNAME: Required. Identify the data set by its cataloged name. If the catalog contains more than one index level, the data set name must be fully qualified.

DISP: Required. Give the data set's status, OLD or SHR, and indicate how you want it treated after its use, unless you want it to remain cataloged.

LABEL: Required only if the data set does not have a standard label.

DCB: Required only if complete data control block information is not specified by the processing program and the data set label. To save recoding time, you can copy DCB attributes from an existing DCB parameter and modify them if necessary. Valid DCB subparameters are listed in Appendix B.

Examples 18 and 19 illustrate valid DD statements for retrieving cataloged data sets.

```
//INPUT2 DD DSNAME=A.B.C,DISP=(OLD,UNCATLG)
```

Example 18. Retrieving and Uncataloging a Data Set

```
//INPUT2 DD DSNAME=X.Y.Z,DISP=SHR
```

Example 19. Retrieving a Disk Data Set, Which Can Be Shared by Another Job

Note: In addition to the disposition UNCATLG, you can pass a cataloged data set to a later step (PASS), or delete it (DELETE).

Retrieving Noncataloged (Kept) Data Sets

Input data sets that were assigned a disposition of KEEP are retrieved by their tabulated name and location, using the DD statement parameters DSNAME, UNIT, VOLUME, DISP, LABEL, and DCB.

DSNAME: Required. Identify the data set by the name assigned to it when it was created.

UNIT: Required, unless VOLUME=REF is used. Identify the unit by its address, type, or group name. If the data set requires more than one unit, give the number of units. You can also request deferred volume mounting and unit separation with this parameter.

VOLUME: Required. Identify the volume or volumes with serial numbers or, if the data set was retrieved earlier in the same job, with VOLUME=REF. If you want the volume to be PRIVATE, specify PRIVATE. If you want a private volume to remain mounted until a later job step uses it, specify RETAIN.

DISP: Required. Give the data set's status, OLD or SHR, and indicate how you want it treated after its use.

LABEL: Required if the data set does not have a standard label. If the data set resides with others on tape, you must give its sequence number.

DCB: Required for all indexed sequential data sets. Otherwise, required only if complete data control block information is not supplied by the processing program and the data set label. To save recoding time, you can copy DCB attributes from an existing DCB parameter, and modify them if necessary. Valid DCB subparameters are listed in Appendix B.

Examples 20 through 22 illustrate valid DD statements for retrieving noncataloged data sets.

```

//INPUT3 DD DSNAME=ALPHA,UNIT=2311,DISP=SHR, X
//          VOLUME=SER=(P12,P14)

```

Example 20. Retrieving a Noncataloged Data Set, Which Can Be Shared by Another Job

```

//INPUT3 DD DSNAME=BETA,UNIT=2400,LABEL=(2,BLP), X
//          DISP=(OLD,DELETE),VOLUME=(PRIVATE,RETAIN,SER=T3), X
//          DCB=(*.STEP1.OUTPUT,DEN=2)

```

Example 21. Retrieving and Deleting a Noncataloged Data Set, With Bypass Label Processing

```

//INPUT DD DSNAME=MHB,DCB=DSORG=IS,UNIT=(2311,3), X
//          DISP=(OLD,KEEP),VOLUME=SER=(334,335,336)

```

Example 22. Retrieving an Indexed Sequential Data Set on Three Disks

Retrieving Passed Data Sets

Input data sets used in a previous job step and passed are retrieved using the DD statement parameters DSNAME, DISP, and UNIT. The data set's unit type, volume location, and label information remain available to the system from the original DD statement.

DSNAME: Required. Identify the original data set by either its name or the DD statement reference term *.stepname.ddname. If the original DD statement occurs in a cataloged procedure, you must include the procedure step name in the reference term.

ADDITIONAL DD STATEMENT FACILITIES

Parameters and fields of the DD statement can be coded in special ways to perform functions other than simply creating and retrieving data sets. Variations of the name field allow you to:

- Concatenate two or more input data sets.
- Use a private library.
- Define data sets used for ABEND dumps.

The DUMMY parameter coded by itself or with other parameters can be used to bypass input/output operations on data sets. The DSNAME parameter, when coded in a special way, can be used in combination with other DD statement parameters to create and retrieve generation data groups. The AFF and SEP parameters allow you to make efficient use of channels in certain situations.

Concatenating Data Sets

Several input data sets, each of which may reside on a different volume, can be read as if they were a single data set through the technique of concatenation. This technique makes it possible for a processing program to get its input from several different types of devices. Concatenated data sets are read in the order of appearance of their DD statements in the input stream.

To concatenate data sets, simply omit the ddnames from all DD statements except the first in sequence. Example 29 illustrates a group of DD statements defining concatenated data sets, including a data set in the input stream.

```
//INPUT    DD  DSNAME=A.B.C,DISP=(OLD,DELETE)
//          DD  DSNAME=X.Y.Z,DISP=OLD,LABEL=(,NL)
//          DD  DSNAME=ALPHA,UNIT=2311,VOLUME=SER=P12,
//          DD  DISP=(OLD,DELETE)
//          DD  *
//          ---Data Cards---
/*
```

Example 29. Concatenating Data Sets

Using a Private Library

Processing programs that are used most frequently reside in the system library, SYS1.LINKLIB. However, you may want to place a program in a private library for one of several reasons:

- It is used infrequently.
- It is not completely checked out.
- It is used only by a limited number of people.
- You wish to transport it from one location to another.

To retrieve a program from a private library, you must first make the library available to a job. The simplest way to do this is by placing a special DD statement at the beginning of the job. When the operating

system encounters this statement, it effectively concatenates the private library with the system library, for the duration of the job. As the job progresses, the system searches for each program, first in the private library, and then in the system library.

The DD statement must contain the special ddname JOBLIB, and must appear immediately after the JOB statement of the job to which it pertains. The operand field, at minimum, must contain the DSNAMES and DISP parameters. The DISP parameter must be coded DISP=(OLD,PASS) or DISP=(SHR,PASS), so that the library remains available throughout the job. (The system assumes DISP=(OLD,PASS) if you code DISP=OLD.) Other parameters should be coded according to requirements for retrieving data sets, as discussed in an earlier chapter.

Example 30 illustrates a valid sequence of control statements for making a private library available to a job.

```

//PAYROLL JOB [JOB statement parameters]
//JOBLIB DD DSNAMES=PRIVATE.LIB1,DISP=(OLD,PASS)
//STEP1 EXEC [EXEC statement parameters]

```

Example 30. Retrieving a Cataloged Private Library

As with ordinary DD statements, you can arrange a sequence of JOBLIB DD statements so that the private libraries they define are effectively read as one. The libraries are searched in the order in which the DD statements appear, with the system library searched last. To concatenate private libraries, omit the ddname from all the DD statements except the first. The first statement must specify a ddname of JOBLIB. The entire group must appear immediately after the JOB statement, and before the first EXEC statement.

Defining Data Sets Used for Abnormal Termination Dumps

Job steps subject to abnormal termination can take advantage of the operating system abnormal termination dumping facilities. To avail a job step of these facilities, you must include a special DD statement defining a data set on which the dump can be written. This DD statement must be identified by one of the special ddnames SYSABEND or SYSUDUMP, and must include appropriate parameters for a basic sequential (BSAM) data set. The processing program must not make a reference to such a data set. If more than one special ddname is included in a job step, all but the first DD statement are ignored.

The dump provided when the SYSABEND DD statement is used includes the system nucleus, the problem program storage area, and a trace table, if the trace table option was requested at system generation. The SYSUDUMP DD statement provides only a dump of the problem program storage area.

If you choose to have the dump routed through the output stream and written on a system output device, you must include the SYSOUT parameter. If you are using a priority scheduler, you can also include the UNIT and SPACE parameters to define the intermediate direct-access device. Appropriate parameters for this type of output are discussed in an earlier topic titled "Routing a Data Set through the Output Stream."

If you choose, instead, to store the dump and write it at a later time, the SYSABEND DD statement must identify the data set, provide unit and volume information, and give a disposition of KEEP or CATLG. If the unit is direct-access, you must also include one of the space allocation parameters. Example 31 illustrates a sample set of job steps that makes use of the abnormal termination dumping facilities.

APPENDIX A: UNIT TYPES

The UNIT parameter of the DD statement can identify an input or output unit by its actual address, its type number, or its group name. Type numbers, automatically established at system generation, correspond to units configured into your system. Type numbers and corresponding units are listed here for your convenience.

Tape Units

<u>Unit Type</u>	<u>Unit</u>
2400	any 2400 Nine-Track Magnetic Tape Drive
2400-1	any 2400 Magnetic Tape Drive with Seven-Track Compatibility
2400-2	2400 Magnetic Tape Drive with Seven-Track Compatibility and Data Conversion
2400-4,-5,-6	any 2400 Nine-Track Magnetic Tape Drive with a density of 1600 bytes per inch. Optional feature allows for a density of 800 bytes per inch. Model number denotes rate of data transmission.

2311	any 2311 Disk Storage Drive
2314	2314 Storage Facility

Unit Record Equipment

<u>Unit Type</u>	<u>Unit</u>
1052	1052 Printer-Keyboards
1053	1053 Printer
1403	1403 Printer or 1404 Printer (continuous form only)
1442	1442 Card Read Punch
1443	any 1443 Printer
2501	2501 Card Reader
2520	2520 Card Read Punch
2540	2540 Card Read Punch (read feed)
2540-2	2540 Card Read Punch (punch feed)
2671	2671 Paper Tape Reader

Graphic Units

2250	2250 Display Unit, Model 1
2250-2	2250 Display Unit, Model 2
2260	2260 Display Station (local attachment)

Direct Access Units

<u>Unit Type</u>	<u>Unit</u>
2301	2301 Drum Storage Unit
2302	2302 Disk Storage Drive
2303	2303 Drum Storage Unit

APPENDIX B: DCB SUBPARAMETERS

The data control block associated with a data set is filled by a number of sources, one of which is the DD statement. The DCB parameter supplies missing or overriding attributes in the form of a list of subparameters. The glossary below lists the keywords that you can code in the DCB parameter, their definitions, and the values they may assume.

Table 3 supplies valid keywords and values that you can use with the indexed sequential, partitioned, direct, and sequential access methods. Subparameters that apply to the graphics access method (GNCP, GDSORG) and the teleprocessing access methods (BUFRQ, CPRI, INTVL, SOWA) do not appear in the table. Further information on DCB subparameters appears in the publication IBM System/360 Operating System: Supervisor and Data Management Macro-Instructions.

Glossary of DCB Subparameters

BFALN	Fullword (F) or doubleword (D) boundary alignment of each buffer.
BFTEK	Type of buffering (simple or exchange) to be supplied by the control program (S or E).
BLKSIZE	Maximum block size in bytes (a number).
BUFL	Length, in bytes, of each buffer to be obtained for a buffer pool (a number).
BUFNO	Number of buffers to be assigned to the data control block.
BUFRQ	Number of buffers to be read in advance from the direct-access device queue. (For use with teleprocessing access methods.)
CODE	Paper tape code in which the data is punched.
	I - IBM BCD perforated tape and transmission code (8 tracks) F - Friden (7 tracks)

B	Burroughs (7 tracks)
C	National Cash Register (8 tracks)
A	ASCII (8 tracks)
T	Teletype (5 tracks)
N	No conversion

CPRI	Relative priority to be given to sending and receiving operations. (For use with teleprocessing access methods.)
------	--

CYLOFL	Number of tracks to be reserved on each cylinder to hold records that overflow from other tracks on that cylinder.
--------	--

DEN	Tape recording density.
	0 - 200 bits/inch (7-track only)
	1 - 556 bits/inch (7-track only)
	2 - 800 bits/inch

For 7-track tapes, all information on the reel must be written in the same density (i.e., labels, data, tapemarks). Do not specify DEN for a SYSOUT data set.

DSORG	Organization of the data set.
-------	-------------------------------

PS	Physical sequential
PSU	Physical sequential unmovable
PO	Partitioned organization
POU	Partitioned organization unmovable
IS	Indexed sequential
ISU	Indexed sequential unmovable
DA	Direct-access
DAU	Direct-access unmovable

EROPT	Option to be executed if an error occurs.
-------	---

ACC	Accept
SKP	Skip
ABE	Abnormal end of task

GDSORG	Organization of a graphic data set. (For use with the graphics access method.)
--------	--

GNCP	Maximum number of input/output macro-instructions that will be issued before a WAIT macro-instruction. (For use with the graphics access method.)
------	---

APPENDIX D: CREATING AND RETRIEVING INDEXED SEQUENTIAL DATA SETS

Indexed sequential (ISAM) data sets are created and retrieved using special subsets of DD statement parameters and subparameters. They can occupy up to three different areas of space:

- Prime area -- This area contains data records and related track indexes. It exists for all ISAM data sets.
- Overflow area -- This area contains overflow from the prime area when new data records are added. It is optional.
- Index area -- This area contains master and cylinder indexes associated with the data set. It exists for any ISAM data set that has a prime area occupying more than one cylinder.

ISAM data sets must reside on direct-access volumes. Because an ISAM data set can be associated with more than one type of unit, it is not usually cataloged.

Creating an ISAM Data Set

ISAM data sets are created with from one to three DD statements. One of the statements must define the prime area. DD statements must define the areas in a specific order:

1. Index area.
2. Prime area.
3. Overflow area.

This order must be maintained if one of the statements is absent. The first or only DD statement defining the data set can contain a name field. Other statements must have a blank name field.

The subset of DD statement parameters used to create an ISAM data set excludes *, DATA, DUMMY, DDNAME, SYSOUT, SUBALLOC, and SPLIT. The remaining DD statement parameters -- DSNAME, UNIT, VOLUME, LABEL, DCB, DISP, SPACE, and SEP and AFF -- are all valid. However, you must follow certain restrictions in using these parameters.

DSNAME: Required. In addition to giving the data set name, the DSNAME parameter identifies the area you are defining, i.e., DSNAME=name(INDEX), DSNAME=name(PRIME), and DSNAME=name(OVFLOW).

Notes:

- If the data set is temporary, replace name with \$name.
- If you are using only one DD statement to define the entire data set, use DSNAME=name(PRIME) or DSNAME=name.

UNIT: Required, unless VOLUME=REF is used. The first subparameter identifies a direct-access unit. If you include separate statements for the prime and index areas, you must request the same number of units for the prime area as there are volumes. You cannot specify DEFER on any of the statements. Another way of requesting units is by using the unit affinity subparameter, AFF.

Notes:

- DD statements for prime and overflow areas must indicate the same type of unit.
- The DD statement for the index area can indicate a unit type different than the others.

VOLUME: Optional. Can be used to request private volumes (PRIVATE), retain private volumes (RETAIN), or to make specific volume references (SER or REF).

LABEL: Optional. Can be used to indicate the use of both standard and user labels (SUL) and to specify a retention period (RETPD or EXPDT) and password protection (PASSWORD).

DCB: Required. Can be used to complete the data control block if it has not been completed by the processing program. You must include in the list of attributes DSORG=IS or DSORG=ISU, even though this attribute was provided in the processing program. If more than one DD statement is used to define the data set, the DCB parameters in the statements must not contain conflicting attributes.

DISP: Optional. Must be coded if you want to keep the data set (KEEP), catalog it (CATLG), or pass it to a later job step (PASS). An ISAM data set can be cataloged using CATLG only if all three areas are defined by the same DD statement.

Note:

- You can catalog ISAM data sets defined by more than one DD statement by using the system utility program IEHPROGM, provided all volumes reside on the same type of unit. The utility program IEHPROGM is described in the publication IBM System/360 Operating System: Utilities.

SPACE: Required. You must request space using either the recommended nonspecific allocation technique or the more restricted absolute track (ABSTR) technique. If more than one DD statement is used to define the data set, all must request space using the same technique.

If you use the nonspecific space allocation technique, space must be requested in units of cylinders (CYL). The quantity of space you request is assigned to the area identified in the DSNAMES parameter. If you requested more than one unit, this quantity of space is allocated on each volume used by the data set. You cannot request incremental space for ISAM data sets. If you are using one DD statement to define both the index and prime areas, you can indicate the size of the index in the SPACE parameter of the DD statement that defines the prime area. The subparameters RLSE, MXIG, ALX, and ROUND cannot be used. You can, however, request contiguous space on each of the volumes occupied by the data set with the subparameter CONTIG. If CONTIG is coded on one of the statements, it must be coded on all of them.

If you use the absolute track technique of allocating space, the number of tracks must be equivalent to an integral number of cylinders. The address of the beginning track must correspond with the first track of a cylinder other than the first cylinder on a volume. If you requested more than one unit, space is allocated beginning at the specified address and continuing through the volume and onto the next volume until the request has been satisfied. If you are using one DD statement to define both the index and prime areas, you can indicate the size of the index (in tracks)

in the SPACE parameter of the DD statement defining the prime area. This number must also be equivalent to an integral number of cylinders.

Notes:

- The first volume to be allocated for the prime area of an ISAM data set cannot be the volume from which the system is loaded (the IPL volume).
- Space can be requested on more than one volume only on the DD statement that defines the prime area.

SEP and AFF: Optional. You can request channel separation from earlier data sets on any of the DD statements in the group. If you wish to have areas of an ISAM data set written using separate channels, you must request units by their actual address, e.g., UNIT=190.

Example 42 illustrates a valid set of DD statements for creating an ISAM data set. In this example, each area is defined by a separate DD statement.

The manner in which the areas of an ISAM data set are arranged is based primarily on two criteria:

1. The number of DD statements used to define the data set.
2. The types of DD statements used (as reflected in the DSNAMES parameter).

An additional criterion arises when you do not include a DD statement for the index area:

3. Is an index size coded in the SPACE parameter of the DD statement defining the prime area?

Table 6 illustrates the arrangements resulting from various permutations of the above criteria. In addition, it points out restrictions on the number and type of units that can be requested for each permutation.

```
//OUTPUT4 DD DSNAMES=MHB(INDEX),UNIT=2301,DCB=DSORG=IS, X|
//          SPACE=(CYL,10,,CONTIG),DISP=(,KEEP)
//
//          DD DSNAMES=MHB(PRIME),DCB=DSORG=IS,UNIT=(2311,2), X|
//          VOLUME=SER=(334,335),DISP=(,KEEP), X|
//          SPACE=(CYL,25,,CONTIG)
//
//          DD DSNAMES=MHB(OVFLOW),DCB=DSORG=IS,UNIT=2311, X|
//          VOLUME=SER=336,SPACE=(CYL,25,,CONTIG),DISP=(,KEEP)
```

Example 42. Creating an Indexed Sequential Data Set

<u>The DSNNAME Parameter</u>		<u>The DISP Parameter</u>	
DSNAME=	$\left\{ \begin{array}{l} \text{name} \\ \text{name(area name)} \\ \text{name(membername)} \\ \text{name(generation \#)} \\ \text{\%name(membername)} \\ \text{\%name(area name)} \\ \text{*.stepname.ddname} \end{array} \right\}$	DISP=	$\left(\begin{array}{l} \text{[SHR]} \\ \text{NEW} \\ \text{OLD} \\ \text{MOD} \end{array} \left[\begin{array}{l} \text{[DELETE]} \\ \text{KEEP} \\ \text{PASS} \\ \text{CATLG} \\ \text{UNCATLG} \end{array} \right] \right)$
<u>The Unit Parameter</u>			
UNIT=	$\left(\left[\begin{array}{l} \text{address} \\ \text{type} \\ \text{group} \end{array} \right] \left[\begin{array}{l} \text{[P]} \\ \text{[n]} \end{array} \right] \left[\text{[DEFER]} \right] \left[\text{[SEP=(list of ddnames)]} \right] \right)$		
<u>The VOLUME Parameter</u>			
VOLUME=	$\left(\left[\text{[PRIVATE]} \right] \left[\text{[RETAIN]} \right] \left[\text{[volseq\#]} \right] \left[\text{[volcount]} \right] \left[\begin{array}{l} \text{[SER=(list of serial \#s)]} \\ \text{[REF=dsname]} \\ \text{[REF=*.stepname.ddname]} \end{array} \right] \right)$		
<u>The LABEL Parameter</u>			
LABEL=	$\left(\left[\text{[data set seq\#]} \right] \left[\begin{array}{l} \text{[SL]} \\ \text{[SUL]} \\ \text{[NSL]} \\ \text{[NL]} \\ \text{[BLP]} \end{array} \right] \left[\text{[PASSWORD]} \right] \left[\begin{array}{l} \text{[EXPDT=yyddd]} \\ \text{[RETPD=nnnn]} \end{array} \right] \right)$		
<u>Space Allocation Parameters</u>			
SPACE=	(ABSTR, (quantity, address[, directory]))		
SPACE=	$\left(\left\{ \begin{array}{l} \text{TRK} \\ \text{CYL} \\ \text{blocksize} \end{array} \right\}, \left(\text{quantity} \left[\text{[increment]} \right] \left[\text{[directory]} \right] \left[\text{[RLSE]} \right] \left[\begin{array}{l} \text{[CONTIG]} \\ \text{[MXIG]} \\ \text{[ALX]} \end{array} \right] \left[\text{[ROUND]} \right] \right) \right)$		
SPLIT=	$\left\{ \begin{array}{l} \text{n} \\ \text{(n, CYL, (quantity[, increment]))} \\ \text{\%} \\ \text{(\%, blocksize, (quantity[, increment]))} \end{array} \right\}$		
SUBALLOC=	$\left(\left\{ \begin{array}{l} \text{TRK} \\ \text{CYL} \\ \text{blocksize} \end{array} \right\}, \left(\text{quantity} \left[\text{[increment]} \right] \left[\text{[directory]} \right] \right), \text{stepname.ddname} \right)$		
<u>Legend:</u>			
{ } Choose one.			
[] Optional; if more than one line is enclosed, choose one or none.			

